

# 帷策公司 Hadoop+Spark

## 目录

一、什么是 hadoop.....	1
二、Spark 与 hadoop 关系.....	2
三、公司 hadoop 配置详情.....	2
四、部署步骤.....	3
步骤一、环境准备.....	3
步骤二、在 Namenode01 主机中修改 hadoop 配置文件，然后复制到其它各主机.....	4
步骤三、在 namenode01 中启动 HDFS 和 YARN.....	9
步骤四、配置 spark.....	11
步骤五、集群操作.....	13
步骤六、节点管理.....	15
一、新增数据节点.....	15
二、删除数据节点.....	18

该文档简单介绍了 hadoop 组成，并用于构建和维护公司的 Hadoop+Spark 集群

## 一、什么是 hadoop

Hadoop 是 Apache 软件基金会支持可靠的、可扩展的一个开源的分布式计算框架的工程。

Apache Hadoop 软件库是一个允许使用简单编程模型跨计算机集群处理大型数据集合的框架，其设计的初衷是将单个服务器扩展成上千个机器组成的一个集群为大数据提供计算服务，其中每个机器都提供本地计算和存储服务

Hadoop 由多个组件构成，核心组件如下：

**Hadoop Common：**支持其他 Hadoop 模块的通用工具。

**Hadoop Distributed File System(HDFS)：**提供高吞吐量的访问应用数据的一个分布式文件系统。

**Hadoop YARN：**一种作业调度和集群资源管理的框架。

**Hadoop MapReduce：**一种基于 Yarn 来处理大数据集合的系统。

**Spark：**一种用于 Hadoop 数据的快速通用计算引擎。Spark 提供一种支持广泛应用的简单而易懂的编程模型，包括 ETL（Extract-Transform-Load）、机

器学习、流处理以及图计算。

除此还有多个组件 Avro、Hive、Hbase、Pig、Zookeeper、Mahout、Casscandra、Tez 等

## 二、Spark 与 hadoop 关系

如果说 MapReduce 是第一代计算引擎，那么 Spark 就是第二代计算引擎。Spark 优点：速度快、易用性、通用性、支持多种模式运行

## 三、公司 hadoop 配置详情

公司 Hadoop 版本 hadoop-2.7.7, spark 版本 spark-2.3.2

Hadoop 服务器列表

服务器 IP	主 机 名	任务列表	启 用 端 口	安装目录	软件
192.168.0.201	Name node 01	NameNode SecondaryNamenode ResourceManager	50070 50090 8088 9000	/opt/hadoop-2.7.7 /opt/spark-2.3.2	HDFS Yarn
192.168.0.205	Data node 03	DataNode NodeManager	50075 8042	/opt/hadoop-2.7.7	HDFS
192.168.0.206	Data node 04	DataNode NodeManager	50075 8042	/opt/hadoop-2.7.7	HDFS
192.168.0.207	Data node 05	DataNode NodeManager	50075 8042	/opt/hadoop-2.7.7	HDFS

## 四、部署步骤

### 步骤一、环境准备

1) hadoop 是用 java 编写的, spark 是由 Scala 编写的, 所以还需要安装 java-1.8.0-openjdk-devel 和 scala, hadoop 需要集群所有设备安装, 即集群所有设备需要安装 jdk 环境, spark 只需要 Namenode01 安装

```
yum -y install java-1.8.0-openjdk-devel    #安装 jdk
```

```
java -version                             #查看版本
```

```
rpm -ql java-1.8.0-openjdk-devel          #查看安装路径
```

剩余软件也可手动下载, 但要注意各版本之间能否兼容, 以下为网址:

JDK: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Hadoop: <http://archive.apache.org/dist/hadoop/common/>

Scala: <https://www.scala-lang.org/download/all.html>

Spark: <http://archive.apache.org/dist/spark/spark-2.2.0/>  
<http://spark.apache.org/>

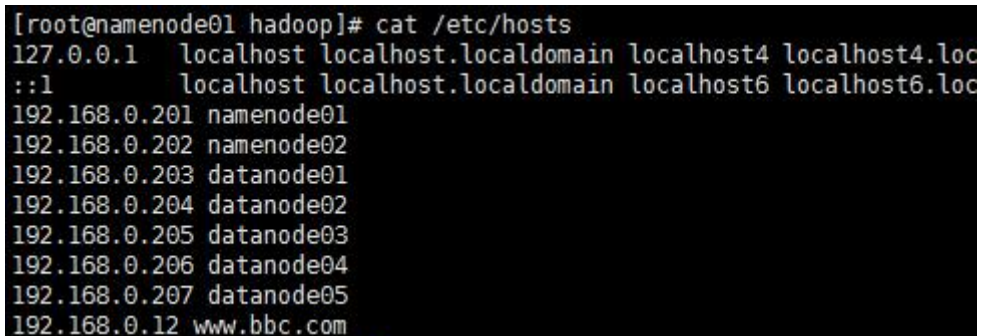
2) Namenode01 与其它主机之间免密登录, 在 Namenode01 运行:

```
ssh-keygen -f /root/.ssh/id_rsa -N ' '    #生成密钥文件
```

```
ssh-copy-id 192.168.0.205/206/207        #传密钥文件到各主机
```

3) 做好各主机的域名解析, 每台主机运行:

```
vim /etc/hosts
```



```
[root@namenode01 hadoop]# cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.0.201 namenode01
192.168.0.202 namenode02
192.168.0.203 datanode01
192.168.0.204 datanode02
192.168.0.205 datanode03
192.168.0.206 datanode04
192.168.0.207 datanode05
192.168.0.12 www.bbc.com
```

\*正式环境还需做 NTP 时间同步以 namenode01 为时间服务器, vim /etc/chrony.conf

步骤二、在 Namenode01 主机中修改 hadoop 配置文件, 然后复制

## 到其它各主机

### 1)、将下载好的压缩包解压至/opt/hadoop-2.7.7

```
tar -xvf /root/hadoop-2.7.7.tar.gz -C /opt/
```

```
[root@namenode01 hadoop]# ls /opt
cloudera hadoop-2.7.7 jdk1.8.0_181 livy-0.5.0 rh spark-2.3.2
```

### 2)、进入配置文件目录:cd /opt/hadoop-2.7.7/etc/hadoop/

配置 hadoop-env.sh, 命令:

```
vim hadoop-env.sh    #在 hadoop-env.sh 中修改 JAVA_HOME
```

```
25     export JAVA_HOME=/usr/local/jdk          # 根据 rpm -ql
java-1.8.0-openjdk-devel 名称查看实际路径
```

```
33     export HADOOP_CONF_DIR="/usr/local/hadoop/etc/hadoop"
```

```
24 # The java implementation to use.
25 export JAVA_HOME=/opt/jdk1.8.0_181
26
27 # The jsvc implementation to use. Jsvc is required to run secure da
28 # that bind to privileged ports to provide authentication of data t
29 # protocol. Jsvc is not required if SASL is configured for authent
30 # data transfer protocol using non-privileged ports.
31 #export JSVC_HOME=${JSVC_HOME}
32
33 export HADOOP_CONF_DIR=${HADOOP_CONF_DIR:-"/etc/hadoop"}
:set nu
```

### 3)、配置 core-site.xml

```
vim core-site.xml
```

#在文件中加入如下内

```
<configuration>
```

```
20     <property>
```

```
21         <name>fs.defaultFS</name>
```

```
22         <value>hdfs://namenode01:9000</value>
```

```
23     </property>
```

```
24     <property>
```

```
25         <name>hadoop.tmp.dir</name>
```

```

26         <value>/data/hadoop277/hadoop</value>
27     </property>
28     <property>
29         <name>io.file.buffer.size</name>
30         <value>65536</value>
31     </property>
</configuration>

```

```

19 <configuration>
20   <property>
21     <name>fs.defaultFS</name>
22     <value>hdfs://namenode01:9000</value>
23   </property>
24   <property>
25     <name>hadoop.tmp.dir</name>
26     <value>/data/hadoop277/hadoop</value>
27   </property>
28   <property>
29     <name>io.file.buffer.size</name>
30     <value>65536</value>
31   </property>
32 </configuration>

```

#### 4) 、修改 hdfs-site.xml

```
vim hdfs-site.xml
```

```

<configuration>
20   <property>
21     <name>dfs.replication</name>
22     <value>3</value>
23   </property>
24   <property>
25     <name>dfs.namenode.secondary.http-address</name>
26     <value>namenode01:50090</value>
27   </property>
28   <property>
29     <name>dfs.namenode.secondary.https-address</name>
30     <value>namenode01:50091</value>

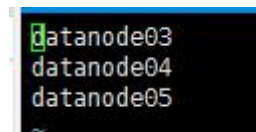
```

```
31     </property>
</configuration>
```

```
19 <configuration>
20   <property>
21     <name>dfs.replication</name>
22     <value>3</value>
23   </property>
24   <property>
25     <name>dfs.namenode.secondary.http-address</name>
26     <value>namenode01:50090</value>
27   </property>
28   <property>
29     <name>dfs.namenode.secondary.https-address</name>
30     <value>namenode01:50091</value>
31   </property>
32 </configuration>
:set nu
```

## 5)、配置 slaves 文件

```
vim slaves
datanode03
datanode04
datanode05  # 添加数据节点的主机名
```



## 6)、配置 yarn-site.xml

```
vim yarn-site.xml
15 <configuration>
16   <property>
17     <name>yarn.resourcemanager.hostname</name>
18     <value>namenode01</value>
19   </property>
20   <property>
21     <name>yarn.nodemanager.aux-services</name>
```

```

22         <value>mapreduce_shuffle</value>
23     </property>
24     <property>
25         <name>yarn.nodemanager.vmem-check-enabled</name>
26         <value>>false</value>
27         <description>Whether virtual memory limits will be enforced
for containers</description>
28     </property>
29     <property>
30         <name>yarn.nodemanager.vmem-pmem-ratio</name>
31         <value>4</value>
32         <description>Ratio between virtual memory to physical memory
when setting memory limits fo    r containers</description>
33     </property>
34 </configuration>

```

```

15 <configuration>
16   <property>
17     <name>yarn.resourcemanager.hostname</name>
18     <value>namenode01</value>
19   </property>
20   <property>
21     <name>yarn.nodemanager.aux-services</name>
22     <value>mapreduce_shuffle</value>
23   </property>
24   <property>
25     <name>yarn.nodemanager.vmem-check-enabled</name>
26     <value>>false</value>
27     <description>Whether virtual memory limits will be enforced for containers</description>
28   </property>
29   <property>
30     <name>yarn.nodemanager.vmem-pmem-ratio</name>
31     <value>4</value>
32     <description>Ratio between virtual memory to physical memory when setting memory limits fo
r containers</description>
33   </property>
34 </configuration>

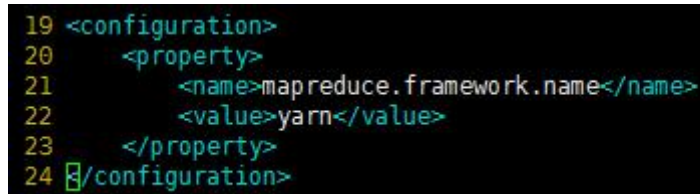
```

## 7) 、配置 mapred-site.xml 文件

该文件是不存在的，把 mapred-site.xml.template 文件改名

```
cp mapred-site.xml.template mapred-site.xml
```

```
vim mapred-site.xml
19 <configuration>
20     <property>
21         <name>mapreduce.framework.name</name>
22         <value>yarn</value>
23     </property>
24 </configuration>
```



```
19 <configuration>
20     <property>
21         <name>mapreduce.framework.name</name>
22         <value>yarn</value>
23     </property>
24 </configuration>
```

8)、配置完以上几个配置文件后，可把配置文件复制到集群其它主机

```
scp -r /opt/hadoop-2.7.7/ datanode03:/opt/hadoop-2.7.7
```

#datanode04 和 datanode05 也是一样

**\*配置完成，接着格式化 hdfs**

**\*在主节点 namenode01 上执行以下命令**

**\*hdfs namenode -format (只能操作一次，不能多次操作，公司集群已操作过，不需操作)**

### 步骤三、在 namenode01 中启动 HDFS 和 YARN

(该步骤在 namenode01 中操作，其它主机查看服务即可)

#### 1)、在 namenode01 中启动 HDFS

201 上有一个进程占用了 9000 端口：



`lsof -i :9000` #查出进程号，然后 kill 掉。

`start-dfs.sh` #启动 hdfs 服务

用命令 `jps` 查看各台机器上对应进程已经起来：

201 主机：NameNode、SecondaryNameNode；

204-207 主机：DataNode

## 2)、启动 Yarn

在 201 上执行：

`start-yarn.sh`

`jps` 查看各台机器上对应进程已经起来：

201 主机新增服务：ResourceManager；

204-207 新增服务：NodeManager；

201 主机最终服务：

```
[root@namenode01 hadoop]# jps
26256 Jps
14226 SecondaryNameNode
14006 NameNode
14410 ResourceManager
[root@namenode01 hadoop]#
```

205、206、207 主机最终服务：

```
[root@datanode04 ~]# jps
28401 Jps
23671 NodeManager
23544 DataNode
[root@datanode04 ~]#
```

使用命令检测集群情况：

`hdfs dfsadmin -report`

```

[root@namenode01 hadoop-2.7.7]# hdfs dfsadmin -report
19/08/13 15:46:51 WARN util.NativeCodeLoader: Unable to load native-hadoop li
. using builtin-java classes where applicable
Configured Capacity: 8258740043776 (7.51 TB)
Present Capacity: 6005691953152 (5.46 TB)
DFS Remaining: 5045080907776 (4.59 TB)
DFS Used: 960611045376 (894.64 GB)
DFS Used%: 16.00%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0

-----
Live datanodes (3):

Name: 192.168.0.207:50010 (datanode05)
Hostname: datanode05
Decommission Status : Normal
Configured Capacity: 2910435999744 (2.65 TB)
DFS Used: 320203591680 (298.21 GB)
Non DFS Used: 27874140160 (25.96 GB)
DFS Remaining: 2562358267904 (2.33 TB)
DFS Used%: 11.00%
DFS Remaining%: 88.04%
Configured Cache Capacity: 0 (0 B)

```

可登陆 hadoop 网站检测集群情况:

201 主机

<http://192.168.0.201:50070/> //--namenode web 页面 (namenode1)

<http://192.168.0.201:50090/> //--secondory namenode web 页面 (namenode1)

<http://192.168.0.201:8088/> //--secondory namenode web 页面 (namenode1)

<http://192.168.0.205:50075/> //--datanode web 页面  
(datanode3, datanode4, datanode5)

<http://192.168.0.205:8042/> //--nodemanager web 页面  
(datanode3, datanode4, datanode5)

<

192.168.0.201:50070/dfshealth.html#tab-overview

度一下,你... 淘宝网 - 淘!... 阿里巴巴热... 湘潭大学 高中物理电子... 慕课网 哔哩哔哩弹幕... 网

Hadoop

Overview

Datanodes

Datanode Volume Failures

Snapshot

Startup Progress

# Overview 'namenode01:9000' (active)

Started:	Mon Aug 12 15:30:31 CST 2019
Version:	2.7.7, rc1aad84bd27cd79c3d1a7dd58202a8c3ee1ed3ac
Compiled:	2018-07-18T22:47Z by stevel from branch-2.7.7
Cluster ID:	CID-a185ea30-ff9c-40ca-9778-0dabd8a04ffe
Block Pool ID:	BP-1010503285-192.168.0.201-1539176083705

# Summary

Security is off.

Safemode is off.

140 files and directories, 2447 blocks = 2587 total filesystem object(s).

Heap Memory used 43.74 MB of 391 MB Heap Memory. Max Heap Memory is 889 MB.

## 步骤四、配置 spark

1)、解压 spark 安装包到/usr/local/目录下

命令:

```
cd /data/soft
```

```
tar -xvf spark-2.2.0-bin-hadoop2.6.tgz -C /usr/local/
```

## 2)、配置 slaves

命令:

```
cd /usr/local/spark-2.2.0-bin-hadoop2.6/conf
```

```
cp slaves.template slaves
```

```
vim slaves
```

写入如下内容

```
Datanode03
```

```
Datanode04
```

```
Datanode05
```

## 3)、配置 spark-env.sh

命令:

```
cp spark-env.sh.template spark-env.sh
```

```
vi spark-env.sh
```

在文件中加入以下内容:

```
export JAVA_HOME=/usr/local/jdk
```

```
export HADOOP_HOME=/usr/local/hadoop-2.6.5
```

```
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
```

```
export SPARK_MASTER_HOST=hadoop1
```

```
export SPARK_MASTER_PORT=7077
```

如下:

```
# - SPARK_NO_DAEMONIZE Run the proposed command in the foreground

export JAVA_HOME=/usr/local/jdk
export HADOOP_HOME=/usr/local/hadoop-2.6.5
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export SPARK_MASTER_HOST=hadoop1
export SPARK_MASTER_PORT=7077
"spark-env.sh" 68L, 3886C
```

## 步骤五、集群操作

集群操作命令，在 namenode01 中执行（其它数据节点执行也可以），命令位置

/opt/hadoop-2.7.7/sbin/\*

/opt/hadoop-2.7.7/bin/\*

```
[root@namenode01 hadoop-2.7.7]# ls /opt/hadoop-2.7.7/bin/
container-executor  hadoop.cmd  hdfs.cmd  mapred.cmd  test-container-executor  yarn.cmd
hadoop             hdfs       mapred    rcc         yarn
[root@namenode01 hadoop-2.7.7]# ls /opt/hadoop-2.7.7/sbin/
distribute-exclude.sh  refresh-namenodes.sh  start-yarn.cmd      stop-yarn.cmd
hadoop-daemon.sh       slaves.sh              start-yarn.sh       stop-yarn.sh
hadoop-daemons.sh     start-all.cmd         stop-all.cmd       stopyarn.sh
hdfs-config.cmd        start-all.sh          stop-all.sh        yarn-daemon.sh
hdfs-config.sh         start-balancer.sh      stop-balancer.sh    yarn-daemons.sh
httpfs.sh              start-dfs.cmd          stop-dfs.cmd
kms.sh                 start-dfs.sh           stop-dfs.sh
mr-jobhistory-daemon.sh start-secure-dns.sh    stop-secure-dns.sh
```

以上命令可以添加进 bash 中，以后就不需要输入绝对路径。

```
vim ~/.bashrc
```

#在最后一行添上:

```
export PATH=/opt/hadoop-2.7.7/sbin:/opt/hadoop2.7.7/bin/:$PATH
```

```
[root@namenode01 hadoop-2.7.7]# echo $PATH
/usr/lib64/qt-3.3/bin:/opt/spark-2.3.2/bin:/opt/spark-2.3.2/sbin:/opt/hadoop-2.7.7/bin:/opt/hadoop-2.7.7/sbin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/opt/jdk1.8.0_181/bin:/root/bin
[root@namenode01 hadoop-2.7.7]#
```

### 1)、服务管理:

start-all.sh    stop-all.sh    #开启和关闭所有集群服务，不推荐使用

start-dfs.sh    stop-dfs.sh    #关闭 hdfs 服务，NameNode、DataNode 和 SecondaryNameNode 会关闭

start-yarn.sh    stop-yarn.sh    #启动和关闭 yarn 服务，Resourcemanager

和 Nodemanager 会关闭

yarn node -list           #查看 nodemanager 运行情况

hdfs dfsadmin -report   #查看 datanode 运行情况

## 2)、集群文件管理:

hadoop fs -[命令] 目录

hdfs dfs -[命令] 目录   #两条命令都是对文件管理, hadoop 命令针对所有系统文件, 使用最广, hdfs 命令只能针对 HDFS 文件, 以下案例以 hadoop 为例, hdfs 命令也可得到类似结果

上传文件至集群: hadoop fs -put words.txt /test/   #把本地的 words.txt 文件上传至集群中的 /test 目录中

```
[root@namenode01 mnt]# cat words.txt
words
[root@namenode01 mnt]# hadoop fs -ls /test
19/08/13 16:12:11 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform
. using builtin-java classes where applicable
[root@namenode01 mnt]# hadoop fs -put words.txt /test
19/08/13 16:12:27 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform
. using builtin-java classes where applicable
[root@namenode01 mnt]# hadoop fs -ls /test
19/08/13 16:12:33 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform
. using builtin-java classes where applicable
Found 1 items
-rw-r--r--  3 root supergroup          6 2019-08-13 16:12 /test/words.txt
[root@namenode01 mnt]#
```

上传集群后可看到 words.txt 文件

查看集群文件内容: hadoop fs -cat /test/words

```
[root@namenode01 mnt]# hadoop fs -cat /test/words.txt
19/08/13 16:15:12 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform
. using builtin-java classes where applicable
words
[root@namenode01 mnt]#
```

从集群下载某个文件, 下载到当前路径:

hadoop fs -get /path/wc/input/words.txt

显示一个文件的末尾

hadoop fs -tail /weblog/test.log

在 hdfs 上创建目录



```
hadoop fs -mkdir -p /aaa/bbb/cc/dd
```

从 hdfs 的一个路径拷贝 hdfs 的另一个路径

```
hadoop fs -cp /aaa/test1.txt /bbb/test2.txt
```

删除目录及其里面内容

```
hadoop fs -rmr /path/wc/input/words.txt
```

与 linux 文件系统中的用法一样，对文件所属权限

```
1) hadoop fs -chmod 666 /hello.txt
```

```
2) hadoop fs -chown someuser:somegrp /hello.txt
```

统计文件系统的可用空间信息

```
hadoop fs -df -h /
```

统计文件夹的大小信息

```
hadoop fs -du -s -h /aaa/*
```

## 步骤六、节点管理

### 一、新增数据节点

- A: 在新增节点修改主机名 datanode02，安装 jdk 环境
- B: Namenode01 节点中设置到新节点的无密码登陆
- C: 在 Namenode01 节点 slaves 文件中添加新节点
- D: 在所有节点/etc/hosts 文件中增加新节点（所有节点保持一致）
- E: 在新节点中执行

```
hadoop-daemon.sh start datanode
```

```
yarn-daemon.sh start nodemanager
```

```
start-balancer.sh      #同步
```

01. [root@hadoop5 ~]# echo datanode02 > /etc/hostname //在 datanode02  
中更改主机名为 datanode02

02. [root@hadoop5 ~]# hostname datanode

03. [root@datanode02 ~]# yum -y install java-1.8.0-openjdk-devel

04. [root@dataanode02 ~]# mkdir /opt/hadoop

05. [root@namenode01 .ssh]# ssh-copy-id 192.168.1.65      #在 namenode01  
中操作

06. [root@namenode01 .ssh]# vim /etc/hosts

07. 192.168.0.201 namenode01

08. 192.168.0.204 node1

09. 192.168.0.205 node2

10. 192.168.0.206 node3

11. 192.168.0.207 node4

12. [root@namenode01 .ssh]# scp /etc/hosts 192.168.1.65:/etc/

13. [root@namenode01 ~]# cd /opt/hadoop-2.7.7/

14. [root@namenode01 hadoop-2.7.7]# vim ./etc/hadoop/slaves

15. datanode02

16. datanode03

17. datanode04

18. datanode05

19. [root@namenode01 hadoop-2.7.7]# for i in {62..65}; do rsync -aSH  
--delete /usr/local/hadoop/

20. \ 192.168.1.\$i:/opt/hadoop-2.7.7/ -e 'ssh' & done //同步配置

25. [root@datanode02 ~]# cd /opt/hadoop-2.7.7/

26. [root@datanode02 hadoop-2.7.7]# ./sbin/hadoop-daemon.sh start  
datanode



```
[root@datanode02 hadoop-2.7.7]# ./sbin/yarn-daemon.sh start nodemanager
```

#在 datanode02 中单独启动，然后用 jps 查看状态

```
(base) [root@datanode02 hadoop-2.7.7]# ./sbin/yarn-daemon.sh start nodemanager
starting nodemanager, logging to /opt/hadoop-2.7.7/logs/yarn-root-nodemanager-datanode02.out
(base) [root@datanode02 hadoop-2.7.7]# jps
16037 Jps
15385 DataNode
15895 NodeManager
(base) [root@datanode02 hadoop-2.7.7]#
```

任务启动后 jps 查看，启动成功

在新的节点上执行：

```
hadoop-daemon.sh start tasktracker
```

均衡之前的 datanode 的 block，这个会非常耗时

1) 如果不 balance，那么 cluster 会把新的数据都存放在新的 node 上，这样会降低 mapred 的工作效率

2) 设置平衡阈值，默认是 10%，值越低各节点越平衡，但消耗时间也更长

```
[root@slave-004 hadoop]# start-balancer.sh -threshold 5
```

3) 设置 balance 的带宽，默认只有 1M/s，修改 hdfs-site.xml

```
<property>
    <name>dfs.balance.bandwidthPerSec</name>
    <value>1048576</value>
</property>
```

设置同步带宽，在新的节点上执行：

```
01. root@datanode02 hadoop-2.7.7]# ./bin/hdfs dfsadmin
```

```
-setBalancerBandwidth 6000000 #6M/s
```

```
02. Balancer bandwidth is set to 60000000
```

```
03. root@datanode02 hadoop-2.7.7]# ./sbin/start-balancer.sh #开始同步
```

在 namenode01 主机中查看节点状态：hdfs dfsadmin -report

```
-----  
Live datanodes (4):
```

```
Name: 192.168.0.207:50010 (datanode05)  
Hostname: datanode05  
Decommission Status : Normal  
Configured Capacity: 291043599744 (2.65 TB)  
DFS Used: 320204959744 (298.21 GB)  
Non DFS Used: 28345593856 (26.40 GB)  
DFS Remaining: 2561885446144 (2.33 TB)  
DFS Used%: 11.00%  
DFS Remaining%: 88.02%  
Configured Cache Capacity: 0 (0 B)  
Cache Used: 0 (0 B)  
Cache Remaining: 0 (0 B)  
Cache Used%: 100.00%  
Cache Remaining%: 0.00%  
Xceivers: 1  
Last contact: Wed Aug 14 10:45:33 CST 2019
```

```
Name: 192.168.0.204:50010 (datanode02)  
Hostname: datanode02  
Decommission Status : Normal  
Configured Capacity: 2680759746560 (2.44 TB)
```

新增的204datanode02节点

## 二、删除数据节点

01. [root@namenode01 hadoop-2.7.7]#  
vim/opt/hadoop-2.7.7/etc/hadoop/slaves
02. #去掉之前添加的 datanode02
03. datanode03
04. datanode04
05. datanode05
06. [root@namenode01 hadoop-2.7.7]# vim  
/opt/hadoop-2.7.7/etc/hadoop/hdfs-site.xml
07. #在此配置文件里面加入下面四行
08. <property>
09. <name>dfs.hosts.exclude</name>
10. <value>/opt/hadoop-2.7.7/etc/hadoop/exclude</value>
11. </property>

12.

13. [root@namenode01 hadoop-2.7.7]# vim

/opt/hadoop-2.7.7/etc/hadoop/exclude datanode02

### 导出数据

01. [root@namenode01 hadoop-2.7.7]# ./bin/hdfs dfsadmin -refreshNodes

02. Refresh nodes successful

03. [root@namenode01 hadoop-2.7.7]# ./bin/hdfs dfsadmin -report #查看

datanode02 显示 Decommissioned

04.

05. Name: 192.168.0.203:50010 (datanode02)

06. Hostname: datanode02

07. Decommission Status : Decommissioned

08. Configured Capacity: 2135949312 (1.99 GB)

09. DFS Used: 4096 (4 KB)

10. Non DFS Used: 1861509120 (1.73 GB)

[root@datanode02 hadoop-2.7.7]# ./sbin/hadoop-daemon.sh stop datanode

//停止 datanode

stopping datanode

[[root@datanode02 hadoop-2.7.7]# ./sbin/yarn-daemon.sh start

nodemanager

//yarn 增加 nodemanager

[root@datanode02 hadoop-2.7.7]# ./sbin/yarn-daemon.sh stop nodemanager

//停止 nodemanager

stopping nodemanager

[root@datanode02 hadoop-2.7.7]# ./bin/yarn node -list

//yarn 查看节点状态，还是有 node4 节点，要过一段时间才会消失

时间：2019.08.13